Temporal Difference Variational Auto-Encoder

TD-VAE

Presented by: Nitin Kumar

TD-VAE

- Use cases
- Introduction
- Modeling
- TD VAE Model
- Intuition
- Experiment
- Conclusion

Use Cases

- Self Driving Cars
- Robot Travel
- Scene Prediction

Introduction

Description:

- Reinforcement learning in partially observed environments
  - Agents need to build a representation of the uncertainty about the world.
  - Agent endowed with memory could learn such a representation implicitly through model-free reinforcement learning.
  - Issues:
    - 1. Reinforcement signal may be too weak to quickly learn.
    - 2. It will not be able to generalize for collection of tasks.
    - 3. Planning step-by-step is not always a cognitively or computationally realistic approach.

Introduction

Problem Statement:

- The model should learn an abstract state representation of the data and be capable of making predictions at the state level, not just the observation level.
- The model should learn a belief state, i.e. a deterministic, coded representation of the filtering posterior of the state given all the observations up to a given time. A belief state contains all the information an agent has about the state of the world and thus about how to act optimally.
- The model should exhibit temporal abstraction, both by making ‘jumpy’ predictions (predictions several time steps into the future), and by being able to learn from temporally separated time points without backpropagating through the entire time interval.

Introduction

Brief Description of Approach:

- First TD-VAE is developed for the sequential, non-jumpy case, by using a modified evidence lower bound (ELBO) for stochastic state space models which relies on jointly training a filtering posterior and a local smoothing posterior.
- Following the intuition given by the sequential TD-VAE, the full TD-VAE model is developed, which learns from temporally extended data by making jumpy predictions into the future.

Modeling

LATENT STATE – SPACE Construction

- Autoregressive models
  - Simplest way to model sequential data (x 1, . . . , x T) is to use the chain rule to decompose the joint sequence likelihood as a product of conditional probabilities, log p(x 1, . . . , x T) = ∑ tlog p(x t| x 1, . . . , x t−1)
  - This formula can be used by combining an RNN which aggregates information from the past (recursively computing an internal state h t = f(h t−1 , x t)) with a conditional generative model which can score the data xt given the context ht
- Issues:
  - They only make predictions in the original observation space, and don’t learn a compressed representation of data.
  - these models tend to be computationally heavy.
  - Model can be computationally unstable at test time since it is trained as a next step model

Modeling

LATENT STATE – SPACE Construction

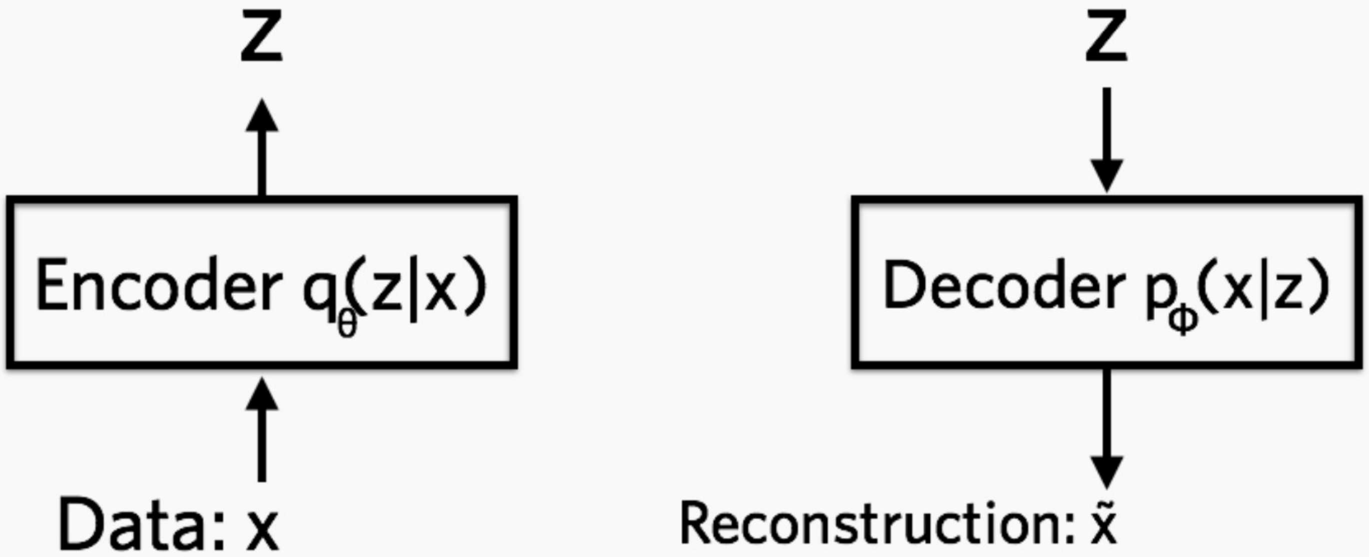
State-space models

- An Alternative to auto regressive model are models which operate on a higher level of abstraction, and use latent variables to model stochastic transitions between states.
- It enables state-to-state transitions only, without needing to render the observations, which can be faster
- They generally consist of decoder or prior networks, which detail the generative process of states and observations, and encoder or posterior networks, which estimate the distribution of latents given the observed data.

Modeling

LATENT STATE – SPACE Construction

State-space models



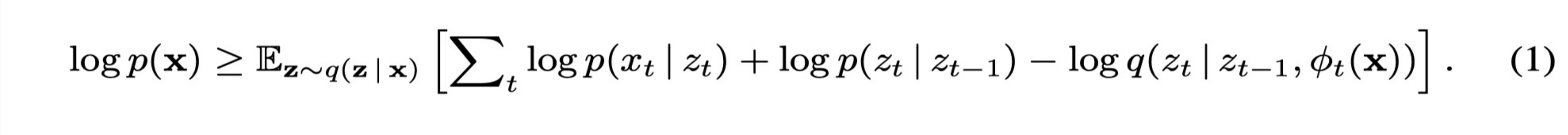
THE ENCODER COMPRESSES DATA INTO A LATENT SPACE (Z). THE DECODER RECONSTRUCTS THE DATA GIVEN THE HIDDEN REPRESENTATION.

Modeling

LATENT STATE – SPACE Construction

State-space models

- Let z = (z 1, . . . , z T) be a state sequence and x = (x 1, . . . , x T) an observation sequence.
- The joint state and observation likelihood can be written as p(x, z) = ∏ tp(z t| z t−1)p(x t| z t)
- These models are commonly trained with a VAEinspired bound, by computing a posterior q(z | x) over the states given the observations.
- Posterior is decomposed autoregressively: q(z | x) = ∏ t q(z t | z t−1 , ϕ t (x)), where ϕ t is a function of (x 1, . . . , x t) for filtering posteriors or the entire sequence x for smoothing posteriors.
- This leads to the following lower bound:



$$\log p(\mathbf{x}) \geq \mathbb{E}_{\mathbf{z}\sim q(\mathbf{z}\,|\,\mathbf{x})}\left[\sum\nolimits_t \log p(x_t\,|\,z_t) + \log p(z_t\,|\,z_{t-1}) - \log q(z_t\,|\,z_{t-1},\phi_t(\mathbf{x}))\right]. \qquad (1)$$

Modeling

Belief State Online creation

- **Sequential models** of data allow to reason about the conditional distribution of the future given the past: p(x t+1, . . . , x T| x 1, . . . , x t)
- b t= b t(x 1, . . . , x t) of the future given the past, which allow to rewrite the conditional distribution as p(x t+1, . . . , x T| x 1, . . . , x t) ≈ p(x t+1, . . . , x T| b t)
- For an autoregressive model as described, the internal RNN state h t can immediately be identified as the desired sufficient statistics b t

Modeling

Belief State Online creation

- For **State space models,** the filtering distribution p(z t| x 1, . . . , x t), also known as the belief state in reinforcement learning, is sufficient to compute the conditional future distribution b t= b t(x 1, . . . , x t) of the future given the past, which allow to rewrite the conditional distribution as p(x t+1, . . . , x T| x 1, . . . , x t) ≈ p(x t+1, . . . , x T| b t)
- Also,

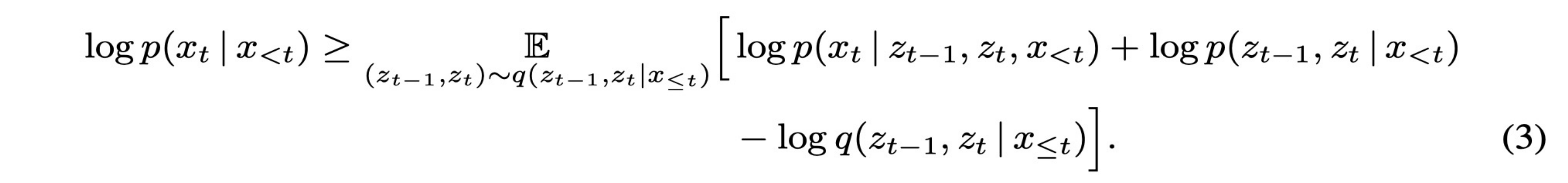
$$p(x_{t+1}, \ldots, x_T \mid x_1, \ldots, x_t) = \int p(z_t \mid x_1, \ldots, x_t) p(x_{t+1}, \ldots, x_T \mid z_t)\, \mathrm{d}z_t. \qquad (2)$$

- if we train a network that extracts a code b t from (x 1, . . . , x t) so that p(z t| x 1, . . . , x t) ≈ p(z t | b t), b t would contain all the information about the state of the world the agent has, and would effectively form a neural belief state.

Modeling

BELIEF - STATE BASED ELBO FOR **SEQUENTIAL** TD-VAE

- Sequential model is developed that satisfies the requirements given in the previous section, namely
  - it constructs a latent state-space
  - it creates a online belief state.
- An arbitrary state space model is considered with joint latent and observable likelihood given by p(x, z) = ∏t p(z t| z t−1)p(x t| z t) and the data likelihood log p(x) is optimized.
- For a given t, we evaluate the conditional likelihood p(x t | x <t) by inferring over two latent states only: z t−1 and z t , as they will naturally make belief states appear for times t − 1 and t:

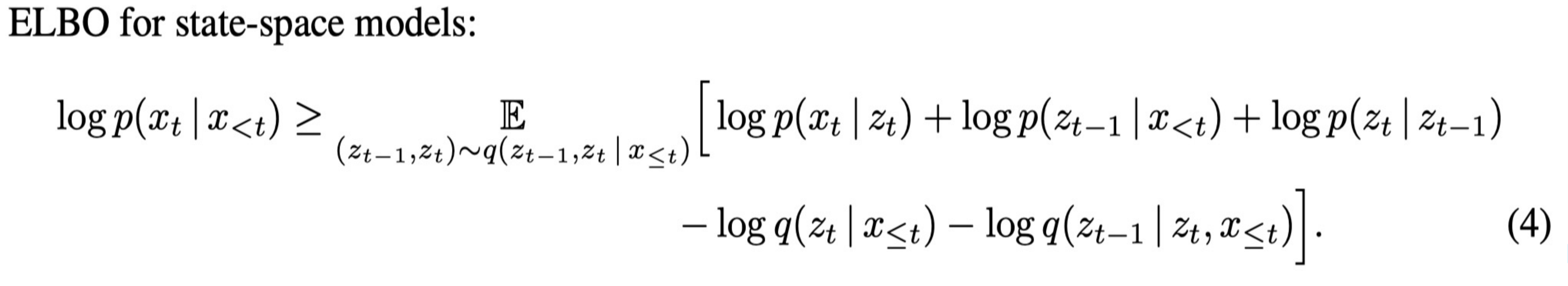


$$\log p(x_t \mid x_{<t}) \geq \mathop{\mathbb{E}}_{(z_{t-1},z_t)\sim q(z_{t-1},z_t \mid x_{\leq t})} \Big[\log p(x_t \mid z_{t-1}, z_t, x_{<t}) + \log p(z_{t-1}, z_t \mid x_{<t}) - \log q(z_{t-1}, z_t \mid x_{\leq t}) \Big]. \quad (3)$$

Modeling

BELIEF - STATE BASED ELBO FOR **SEQUENTIAL** TD-VAE

- We can simplify p(x t| z t−1, z t, x <t) = p(x t| z t) and decompose p(z t−1, z t| x <t) = p(z t−1| x <t)p(z t| z t−1)
- We can decompose q(z t−1, z t| x ≤t) as a belief over z tand a one-step smoothing distribution over z t−1: q(z t−1, z t| x ≤t) = q(z t| x ≤t)q(z t−1| z t, x ≤t)



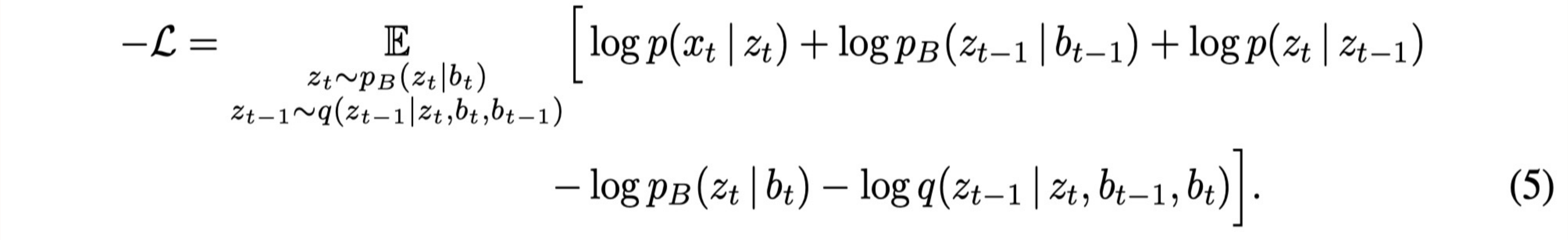
ELBO for state-space models:

$$\log p(x_t \mid x_{<t}) \geq \mathop{\mathbb{E}}_{(z_{t-1},z_t)\sim q(z_{t-1},z_t \mid x_{\leq t})} \Big[\log p(x_t \mid z_t) + \log p(z_{t-1} \mid x_{<t}) + \log p(z_t \mid z_{t-1}) - \log q(z_t \mid x_{\leq t}) - \log q(z_{t-1} \mid z_t, x_{\leq t}) \Big]. \tag{4}$$

Modeling

BELIEF - STATE BASED ELBO FOR **SEQUENTIAL** TD-VAE

- Both quantities p(z t−1 | x ≤ t−1) and q(z t | x ≤ t) represent the belief state of the model at different times, so at this stage we approximate them with the same distribution p B (z | b), with b t = f(b t−1 , x t) representing the belief state code for z t .
- Similarly, we represent the smoothing posterior over z t−1 as q(z t−1| z t, b t−1, b t).



$$-\mathcal{L} = \mathop{\mathbb{E}}_{\substack{z_t \sim p_B(z_t|b_t) \\ z_{t-1} \sim q(z_{t-1}|z_t, b_t, b_{t-1})}} \Big[\log p(x_t \,|\, z_t) + \log p_B(z_{t-1} \,|\, b_{t-1}) + \log p(z_t \,|\, z_{t-1}) - \log p_B(z_t \,|\, b_t) - \log q(z_{t-1} \,|\, z_t, b_{t-1}, b_t) \Big]. \qquad (5)$$

Modeling

TD-VAE AND JUMPY STATE MODELING

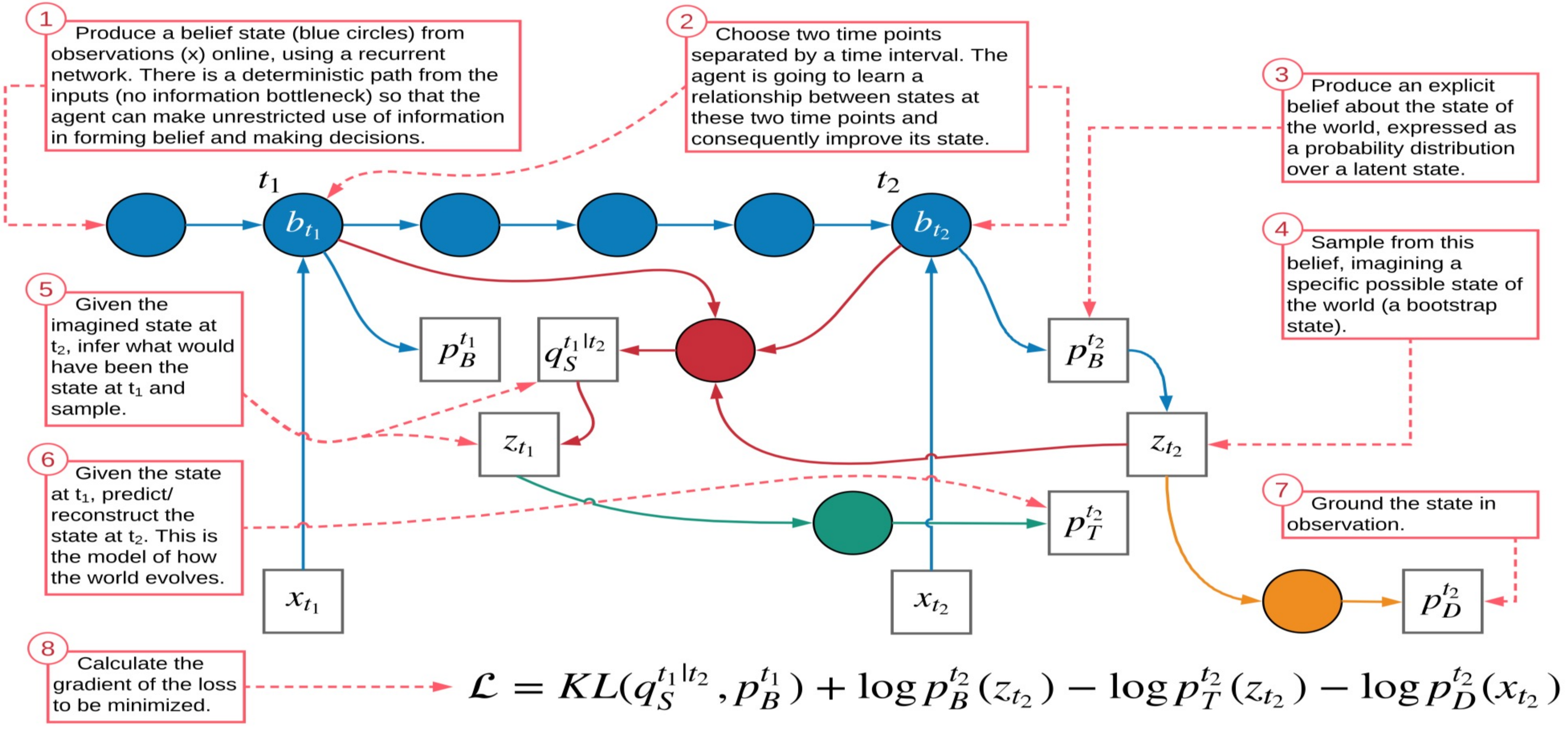
- In many applications, the relevant timescale for planning may not be the one at which we receive observations and execute simple actions.
- for example planning for a trip abroad; the different steps involved (discussing travel options, choosing a destination, buying a ticket, packing a suitcase, going to the airport, and so on), all occur at vastly different time scales (potentially months in the future at the beginning of the trip, and days during the trip).
- Making a plan for this situation does not involve making second-by-second decisions. This suggests that we should look for models that can imagine future states directly, without going through all intermediate states

Modeling

WHY JUMPY STATE MODELING ?

- Training signal coming from the future can be stronger than small changes happening between time steps.
- The behavior of the model should ideally be independent from the underlying temporal sub-sampling of the data, if the latter is an arbitrary choice.
- Jumpy predictions can be computationally efficient.

TD-VAE Model



1 Produce a belief state (blue circles) from observations (x) online, using a recurrent network. There is a deterministic path from the inputs (no information bottleneck) so that the agent can make unrestricted use of information in forming belief and making decisions.

2 Choose two time points separated by a time interval. The agent is going to learn a relationship between states at these two time points and consequently improve its state.

3 Produce an explicit belief about the state of the world, expressed as a probability distribution over a latent state.

4 Sample from this belief, imagining a specific possible state of the world (a bootstrap state).

5 Given the imagined state at t₂, infer what would have been the state at t₁ and sample.

6 Given the state at t₁, predict/ reconstruct the state at t₂. This is the model of how the world evolves.

7 Ground the state in observation.

8 Calculate the gradient of the loss to be minimized.

$$\mathcal{L} = KL(q_S^{t_1|t_2}, p_B^{t_1}) + \log p_B^{t_2}(z_{t_2}) - \log p_T^{t_2}(z_{t_2}) - \log p_D^{t_2}(x_{t_2})$$

● Belief network (filtering)

● Inference network (smoothing)

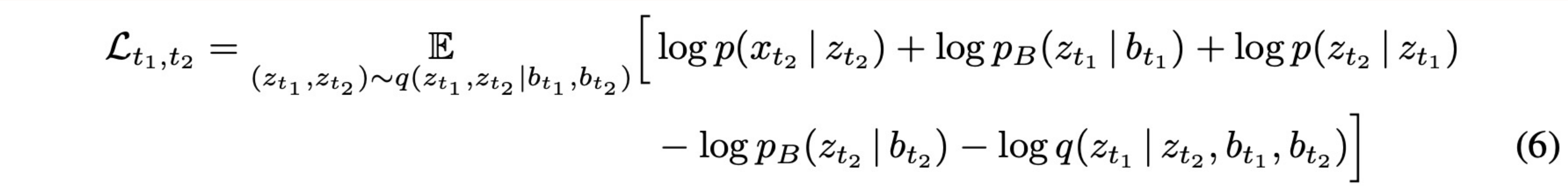
● State prediction network (forward model)

● Decoder network (observation model)

TD-VAE Model

Description

- There exists a sequence of states z1, . . . , zT from which we can predict observations x 1, . . . , x T.
- A forward RNN encodes a belief state bt from past observations x ≤t.
- A jumpy, state-to-state model p(z t2| z t1) between zt1 and z t2 is learnt, and negative loss for TD-VAE is as follows:



$$\mathcal{L}_{t_1,t_2} = \mathop{\mathbb{E}}_{(z_{t_1},z_{t_2})\sim q(z_{t_1},z_{t_2}|b_{t_1},b_{t_2})}\Big[\log p(x_{t_2}\,|\,z_{t_2}) + \log p_B(z_{t_1}\,|\,b_{t_1}) + \log p(z_{t_2}\,|\,z_{t_1}) \\ - \log p_B(z_{t_2}\,|\,b_{t_2}) - \log q(z_{t_1}\,|\,z_{t_2}, b_{t_1}, b_{t_2})\Big] \qquad (6)$$

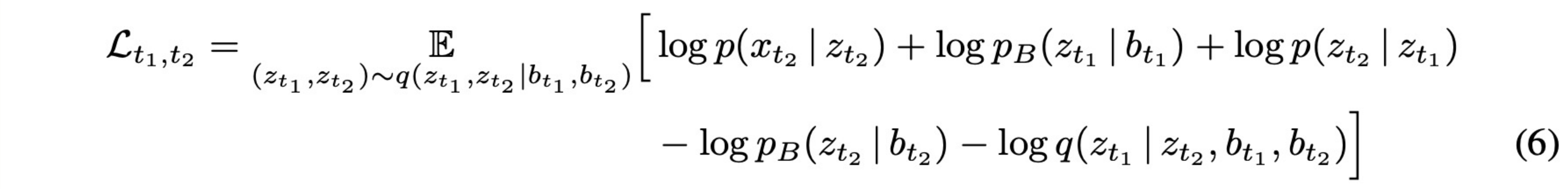
- Training: The distribution of times t 1, t 2 is choosen. Ex. t 1can be chosen uniformly from the sequence, and t 2− t 1uniformly over some finite range [1, D].

Intuition

$$\mathcal{L}_{t_1,t_2} = \mathop{\mathbb{E}}_{(z_{t_1},z_{t_2})\sim q(z_{t_1},z_{t_2}|b_{t_1},b_{t_2})}\Big[\log p(x_{t_2} \mid z_{t_2}) + \log p_B(z_{t_1} \mid b_{t_1}) + \log p(z_{t_2} \mid z_{t_1}) - \log p_B(z_{t_2} \mid b_{t_2}) - \log q(z_{t_1} \mid z_{t_2}, b_{t_1}, b_{t_2})\Big] \qquad (6)$$

- We want to predict a future time step t 2 from all the information we have up until time t 1.
- All relevant information up until time t 1(respectively t 2) has been compressed into a code b t1(respectively b t2).
- We make an observation x t of the world at every time step t, but posit the existence of a state z t which fully captures the full condition of the world at time t.
- At that time, the agent can make a guess of what the state of the world is by sampling from its belief model p B(z t2| b t2).
- Because the state z t2should entail the corresponding observation x t2, the agent aims to maximize p(x t2| z t2) (first term of the loss)
- with a variational bottleneck penalty − log p(z t 2 | b t 2) (second term of the loss) to prevent too much information from the current observation x t2from being encoded into z t
- could the state of the world at time t 2have been predicted from the state of the world at time t 1?

Intuition



$$\mathcal{L}_{t_1,t_2} = \mathop{\mathbb{E}}_{(z_{t_1},z_{t_2})\sim q(z_{t_1},z_{t_2}|b_{t_1},b_{t_2})}\Big[\log p(x_{t_2}\,|\,z_{t_2}) + \log p_B(z_{t_1}\,|\,b_{t_1}) + \log p(z_{t_2}\,|\,z_{t_1}) - \log p_B(z_{t_2}\,|\,b_{t_2}) - \log q(z_{t_1}\,|\,z_{t_2},b_{t_1},b_{t_2})\Big] \tag{6}$$

- By time t 2, the agent has aggregated observations between t 1 and t 2 that are informative about the state of the world at time t 1 , which, together with the current guess of the state of the world z t 2 , can be used to form a guess of the state of the world.
- This is done by computing a smoothing distribution q(z t1| z t2, b t1, b t2) and drawing a corresponding sample z t1.
- Having guessed states of the world zt1 and z t 2 , the agent optimizes its predictive jumpy model of the world state p(z t 2 | z t 1) (third term of the loss)
- it should attempt to assess whether the smoothing distribution q(z t1| z t2, b t2) could have been predicted from information only available at time t 1 (this is indirectly predicting z t 2 from the state of knowledge bt 1 at time t 1- the problem we started with)
- The agent can do so by minimizing the KL between the smoothing distribution and the belief distribution at time t 1 : KL(q(z t 1 | z t 2 , b t 1 , b t 2) || p(z t 1 | b t 1)) (fourth term of the loss)

Experiment

DEEP MIND LAB ENVIRONMENT

- Sequences of frames seen by an agent are used in solving tasks in the DeepMind Lab environment.
- Aim: The model holds explicit beliefs about various possible future and it can roll out in jumps.

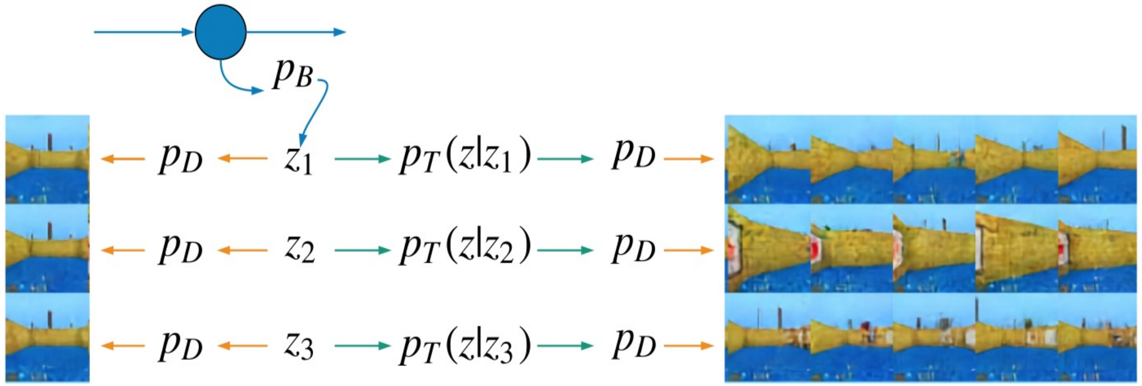


Figure 5: **Beliefs of the model**. **Left**: Independent samples z_1, z_2, z_3 from current belief; all 3 decode to roughly the same frame. **Right**: Multiple predicted futures for each sample. The frames are similar for each z_i, but different across z_i's.

Experiment

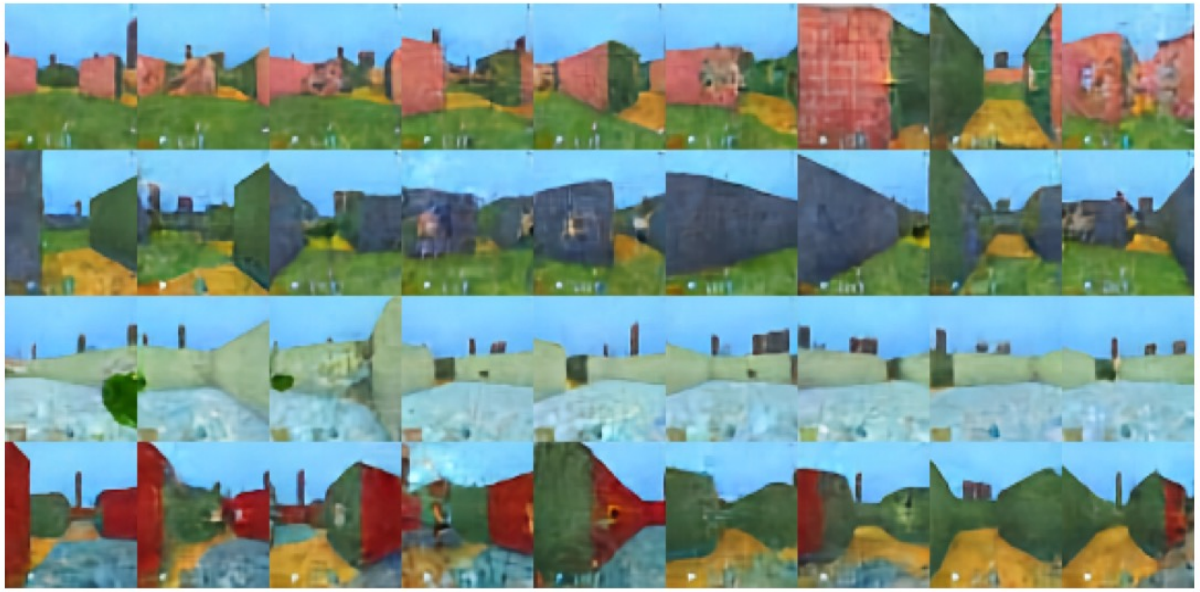


Figure 6: **Rollout from the model**. The model was trained on steps uniformly distributed in $[1, 5]$. The model is able to create forward motion that skips several time steps.

Conclusion

- TD-VAE builds states from observations by bridging time points separated by random intervals.
- Possible Future Work:
  - Creation/Modeling of more generic and accurate state space (since VAE has limitations for the same).

- Extensions:
  - Applications of TD-VAE in more complex setting.
  - Use in Representation learning and planning.

Reference

- Temporal Difference Variational Auto Encoder (TD-VAE)

  - Karol Gregor, George Papamakarios, Frederic Besse, Lars Buesing, Théophane Weber (DeepMind)

  - Published as a conference paper at ICLR 2019

Thank You.